

Particle swarms for numerical wave equation

Michael P. Lamoureux* and Heather K. Hardeman*

ABSTRACT

Motivated by the possibility of computation speed-ups using massive parallelization on fast graphical processing units, we investigate the use of particle swarms to produce a numerical simulation of seismic wave motion in heterogeneous media in 2D and 3D.

It is well known that Brownian motion of particles bouncing about at random forms a useful model for diffusion of heat. In the limit as particle numbers go to infinity, the Brownian motion leads to the diffusion equation: a second order, linear, parabolic partial differential equation. A similar model with correlated, but still random, particle motion leads to the acoustic wave equation in dimensions one, two and three.

Focusing on the Green's function for individual source and receiver pairs in a seismic experiment, we aim to compute the numerical simulation of wave motion using large numbers of independently acting particles to recover the source/receiver response without modelling the entire seismic waveform in the experiment. We present the mathematics behind the theory of the particle simulation as well as a few numerical studies.

INTRODUCTION

A significant computational challenge that arises in seismic imaging is creating a numerical simulation of seismic waves propagating through a complex, three dimensional media. Whether one uses finite difference methods, finite elements or Galerkin methods, or pseudospectral and Fourier transform methods, there is typically a computational grid where all the wavefields are computed, which can become very large even for modest problems. For instance, with a 3D grid of 1000 samples points in each spatial dimension, one immediately obtains one billion grid points ($1000 \times 1000 \times 1000 = 10^9$), each of which becomes a locus for computation even for a simple wavefront traveling through the grid. Each grid point needs to “communicate” at least with its nearest neighbours when computing differential operators, hence the computational burden is increased by the tight intermingling of data at diverse data storage points in memory.

However, there are other models for computation worth considering. A very familiar physical model of Brownian motion involves tracking a large number of particles moving randomly in a medium, resulting in a diffusion process that is accurately described by the heat equation. (Indeed, these ideas go back to Einstein's 1905 work on Brownian motion.) Some useful references on these models of the heat equation appear in Kozdron (2008) and Lawler (2010). More recently, researchers have been using random particle motion as models for a wide variety of stochastic processes that extend such models to equations beyond the heat equation. In the works of Crisan and Lyons (1999), Quer-Sardanyons and Tindel (2007), Toomey and Bean (2000), and Yang and Li (2015) we see these stochastic

*University of Calgary

models applied to the wave equation, other hyperbolic equations, and more exotic partial differential equations. Beyond Brownian motion, one can also consider super Brownian motion (See Slade (2002)) where the randomly moving particles may also undergo branching processes where new particles may be born, and old ones may die. This allows for an even richer class of modelling processes.

In this work, we investigate the use of computational swarms of particles moving under some random process to compute the Green's function for specific instances of the acoustic wave equation in difference dimensions. The idea is that once we have the Green's function for a given source/receiver pair, we have enough information to compute the system response to a source input into the wave medium. (As is done in, for example, Polimeridis et al. (2007).)

A key motivation is that we observe the particles in the swarm move independently of each other, so such a simulation is potentially amenable to parallelization in the computation. Each core in a CPU cluster or GPU assembly of graphics processors can compute independently the paths of its own particles – no communication is required between separate paths. Although we did not have time to run these computational experiments, the potential for large speedups is there.

GREEN'S FUNCTIONS

Let's compute the Green's function for the wave equation in d dimensions, getting the physical units correct.

Start with the constant coefficient wave equation

$$u_{tt} = c^2 \nabla^2 u \quad \text{with initial condition} \quad (1)$$

$$u(\vec{r}, t = 0) = u_0(\vec{r}) \quad (2)$$

$$u_t(\vec{r}, t = 0) = v_0(\vec{r}). \quad (3)$$

The quantity $u(\vec{r}, t)$ is measured in some physical units $[U]$ (displacement, or pressure), position \vec{r} measured in units of length $[L]$ and time t is measured in units of time $[T]$. The coefficient c is a velocity, measured in units of length over time $[L/T]$. The initial conditions will have u_0 measured in units $[U]$ and v_0 in units $[U/T]$.

With constant coefficients in the PDE, the general solution is expressed in terms of two Green's functions $G^{(0)}$ and $G^{(1)}$ by the convolutional formulas

$$u(\vec{r}, t) = \int G^{(0)}(\vec{r} - \vec{r}', t) u_0(\vec{r}') d\vec{r}' + \int G^{(1)}(\vec{r} - \vec{r}', t) v_0(\vec{r}') d\vec{r}'. \quad (4)$$

Matching units, we observe that $G^{(0)}$ is in units of $[1/L^d]$ while $G^{(1)}$ is in units $[T/L^d]$. This is consistent with the observation noted below, that

$$\frac{\partial G^{(1)}}{\partial t} = G^{(0)}, \quad (5)$$

where both sides of this equation are in units $[1/L^d]$.

Both Green's functions satisfy the wave equation

$$G_{tt} = c^2 \nabla^2 G \quad (6)$$

with initial conditions

$$G^{(0)}(\vec{r}, 0) = \delta(\vec{r}) \quad G_t^{(0)}(\vec{r}, 0) = 0 \quad (7)$$

$$G^{(1)}(\vec{r}, 0) = 0 \quad G_t^{(1)}(\vec{r}, 0) = \delta(\vec{r}). \quad (8)$$

To find G , we expand in a Fourier transform, with

$$G(\vec{r}, t) = \int g(\vec{k}, t) e^{2\pi i \vec{k} \cdot \vec{r}} d\vec{k} \quad (9)$$

and observe from the wave equation that the transform g satisfies

$$\ddot{g} = -4\pi^2 k^2 c^2 g \quad (10)$$

which has solution in the general form

$$g(\vec{k}, t) = A(\vec{k}) \cos(2\pi kct) + B(\vec{k}) \sin(2\pi kct). \quad (11)$$

Recalling the delta function identity

$$\delta(\vec{r}) = \int e^{2\pi i \vec{k} \cdot \vec{r}} d\vec{k}, \quad (12)$$

one obtains the initial condition for the transform functions

$$g^{(0)}(\vec{k}, 0) = 1 \quad g_t^{(0)}(\vec{k}, 0) = 0 \quad (13)$$

$$g^{(1)}(\vec{k}, 0) = 0 \quad g_t^{(1)}(\vec{k}, 0) = 1. \quad (14)$$

Solving for $A(\vec{k})$, $B(\vec{k})$ as the constants 0 and 1 gives the two transform functions

$$g^{(0)}(\vec{k}, t) = \cos(2\pi kct) \quad (15)$$

$$g^{(1)}(\vec{k}, t) = \frac{1}{2\pi kc} \sin(2\pi kct). \quad (16)$$

Observe that

$$\frac{\partial}{\partial t} g^{(1)} = g^{(0)} \quad (17)$$

from which we conclude a relation between the Green's functions,

$$\frac{\partial}{\partial t} G^{(1)} = G^{(0)}. \quad (18)$$

In particular, it is enough to find $G^{(1)}$ as the second Green's function can be computed as its derivative.

The formula, then, for the Green's function is given by the Fourier transform of g ; noting the even symmetry, we can include only the real part of the transform, so

$$G^{(1)}(\vec{r}, t) = \int \cos(2\pi \vec{k} \cdot \vec{r}) \frac{\sin(2\pi kct)}{2\pi kc} d\vec{k}. \quad (19)$$

This formula is valid for any dimension d .

THE DIMENSIONS $D = 1, 2, 3$

In dimension $d = 1$, Equation 19 reduces to a single integral

$$G^{(1)}(r, t) = \int_{-\infty}^{\infty} \cos(2\pi kr) \frac{\sin(2\pi kct)}{2\pi kc} dk. \quad (20)$$

which we recognize as the Fourier transform of a sinc function, which can be computed directly. Indeed,

$$G^{(1)}(r, t) = \int_{-\infty}^{\infty} \cos(2\pi kr) \frac{\sin(2\pi kct)}{2\pi kct} t dk = \int_{-\infty}^{\infty} e^{2\pi ikr} \operatorname{sinc}(2kct) t dk. \quad (21)$$

The change of variables $x/2c = tk$, $dx/2c = t dk$ yields

$$G^{(1)}(r, t) = \frac{1}{2c} \int_{-\infty}^{\infty} e^{2\pi ix(\frac{r}{2ct})} \operatorname{sinc}(x) dx \quad (22)$$

$$= \begin{cases} 1/2c, & \text{if } -1/2 < \frac{r}{2ct} \leq 1/2 \\ 0, & \text{otherwise.} \end{cases} \quad (23)$$

Or, in the more familiar form of the light cone, we write

$$G^{(1)}(r, t) = \begin{cases} 1/2c, & \text{if } -ct < r \leq ct \\ 0, & \text{otherwise.} \end{cases} \quad (24)$$

There is a conservation law revealed by this formula, namely that

$$\int_{-\infty}^{\infty} G^{(1)}(r, t) dr = t, \text{ for } t > 0, \quad (25)$$

obtained by integrating the above boxcar integral over the interval $[-ct, ct]$. Since $G^{(0)}$ is the time derivative of $G^{(1)}$, we have

$$\int_{-\infty}^{\infty} G^{(0)}(r, t) dr = 1, \text{ for } t > 0, \quad (26)$$

which suggests we can think of the Green's function $G^{(0)}(r, t)$ as a probability measure over space \mathbb{R}^1 that evolves over time, conserving the total "mass" of the measure.

For dimension $d = 2$, an integration in polar coordinates reveals

$$G^{(1)}(\vec{r}, t) = \begin{cases} \frac{1}{2\pi c} \frac{1}{\sqrt{c^2 t^2 - r^2}}, & \text{for } t > r \\ 0, & \text{otherwise,} \end{cases} \quad (27)$$

while in dimension $d = 3$,

$$G^{(1)}(\vec{r}, t) = \frac{1}{4\pi cr} \delta(ct - r). \quad (28)$$

In all cases, again we have

$$\int_{\mathbb{R}^d} G^{(1)}(\vec{r}, t) d\vec{r} = t, \text{ for } t > 0, \quad (29)$$

and thus

$$\int_{\mathbb{R}^d} G^{(0)}(\vec{r}, t) d\vec{r} = 1, \text{ for } t > 0, \quad (30)$$

We end this section by noting that in dimension 1, 2 and 3, the probability measure $\rho(t) = G^{(0)}(\vec{r}, t)$ is concentrated on a very thin set. Which is why we think the particle swarm would be useful – the particles only need to track this thin wavefront.

BROWNIAN MOTION AND THE HEAT EQUATION

A useful model for the diffusion of heat can be constructed via a large collection of particles that move in space at random (c.f. Zauderer (1989), Lawler (2010)), an idea that Einstein used in his 1905 paper on the analysis of Brownian motion. This is a simpler model than the one required for the wave equation, so it is instructive to analyze this situation first.

Following Zauderer (1989), consider a particle moving at random in one dimension. We assume the particle moves in random jumps of size $\pm\delta$ in time increments τ . Writing X_n for the position of the particle after n time steps, and $x_i = \pm\delta$ the step at the i -th jump, we can write

$$X_n = x_1 + x_2 + \cdots + x_n, \quad (31)$$

from which we can compute the expected value (mean) as

$$E(X_n) = \sum_{i=1}^n E(x_i) = \sum_{i=1}^n (p - q)\delta = (p - q)n\delta, \quad (32)$$

where p, q are the probabilities of the particle moving to the right ($x_i = +\delta$), or to the left ($x_i = -\delta$), respectively.

The expected value of x_i^2 is computed as

$$E(x_i^2) = (+\delta)^2 P(x_i = \delta) + (-\delta)^2 P(x_i = -\delta) = \delta^2(p + q) = \delta^2, \quad (33)$$

since the probabilities p, q must add up to one. We then find the variance of X_n to be

$$V(X_n) = \sum_{i=1}^n V(x_i) = \sum_{i=1}^n [E(x_i^2) - E(x_i)^2] = n(\delta^2 - (p - q)^2\delta^2) = 4pqn\delta^2, \quad (34)$$

where we use the fact that the x_i are independent random variables to expand the variance as a sum.

In order to obtain a continuous flow of heat, we need to set up parameters so that the expectation $E(X_n)$ and variance $V(X_n)$ lead to finite limits as $\delta \rightarrow 0$ and $n \rightarrow \infty$. That is, in the limit, we expect the heat to flow only a finite distance in finite time, and diffuse into

a finite sized region. Examining equations 32 and 34, its clear this cannot happen if p, q are constants, so an easy fix is to allow them to depend on step size, such as in the form

$$p = \frac{1}{2}(1 + b\delta), \quad q = \frac{1}{2}(1 - b\delta). \quad (35)$$

In this case, we have

$$E(X_n) = (p - q)n\delta = bn\delta^2, \quad V(X_n) = 4pqn\delta^2 = (1 - b^2)n\delta^2, \quad (36)$$

thus to get finite limits we simply require that $n\delta^2$ tend to a finite limit as $\delta \rightarrow 0$ and $n \rightarrow \infty$. For a finite time $t = n\tau$, we also require that τ go to zero as n goes to infinite, and thus τ must go to zero at the same rate as δ^2 . That is, we require asymptotic behaviour like

$$\delta \propto \frac{1}{\sqrt{n}}, \quad \tau \propto \frac{1}{n}. \quad (37)$$

To obtain the diffusion equation, define $v(x, t)$ to be the probability that the particle reaches point $x = n\delta$ at time $t = n\tau$. At the next time step, the function $v(x, t + dt)$ will be determined by the density of particles moving in from the left or right, so we get a difference equation

$$v(x, t + \tau) = pv(x - \delta, t) + qv(x + \delta, t). \quad (38)$$

Expanding in a Taylor series in δ and τ , and keeping only orders up to τ^1 and δ^2 (which are similar size), this equation is approximately

$$v(x, t) + v_t(x, t)\tau = p[v(x, t) - v_x(x, t)\delta - \frac{1}{2}v_{xx}(x, t)\delta^2] + q[v(x, t) + v_x(x, t)\delta + \frac{1}{2}v_{xx}(x, t)\delta^2]. \quad (39)$$

Cancelling out the terms $v(x, t) = (p + q)v(x, t)$ and dividing by τ we obtain

$$v_t(x, t) = [(q - p)\frac{\delta}{\tau}]v_x(x, t) + \frac{1}{2}[\frac{\delta^2}{\tau}]v_{xx}(x, t), \quad (40)$$

and recalling that $p, q = 1/2 \pm b\delta$, we have

$$v_t(x, t) = -b[\frac{\delta^2}{\tau}]v_x(x, t) + \frac{1}{2}[\frac{\delta^2}{\tau}]v_{xx}(x, t), \quad (41)$$

In the limit as $n \rightarrow \infty$, the asymptotic choice for δ, τ has the factor δ^2/τ heading to some finite limit, and thus we obtain in the limit a diffusion equation

$$v_t(x, t) = -cv_x(x, t) + \frac{1}{2}Dv_{xx}(x, t) \quad (42)$$

for constants c, D . It is interesting to note that c can be interpreted as a drift coefficient, and arises from the lack of symmetry in that random motion choosing to move left or right at each step. The coefficient D is the diffusion coefficient, and is a measure of how quickly the “heat” diffuses in space.

NUMERICAL VERIFICATION: HEAT EQUATION

Our key observation is that a simulation with random particles should give a useful computational method for finding a Green's function.

As a demonstration, first we show the results of a one particle simulation of random motion, with step size $\delta = 0.1$ and 1000 steps. The code consists of a simple for loop, as follows:

```
x = zeros(1001)
for k=1:1000
    x[k+1] = x[k] + 0.1*(2*rand(0:1) -1)
end
```

Note the factor $(2 * \text{rand}(0 : 1) - 1)$ simply picks at random the direction ± 1 at each step in the loop.

The result is shown in Figure 1, showing the path of the randomly moving particle.

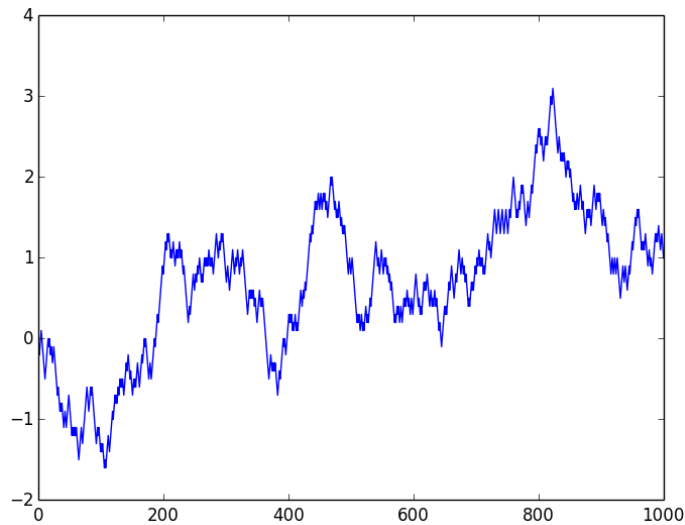


FIG. 1. A simple path in Brownian motion.

To obtain the Green's function, we run a similar loop but with a large number of particles (here, $n_{\text{particles}} = 100,000$) for a large number of steps ($n_{\text{steps}} = 1000/(0.1)^2 = 100,000$) corresponding to spatial step $\delta = 0.1$, time step $\tau = 0.01$ and terminal time $t = 1000$. The core loop is as follows:

```
x = zeros(nparticles)
for k=1:nsteps
    x = x + 0.1*(2*rand(0:1,nparticles)-1 )
end
```

The histogram of particle positions is shown in Figure 2, which is a good approximation to a Gaussian of width $\sqrt{2t} = 44.7$, which is the desired Green's function.

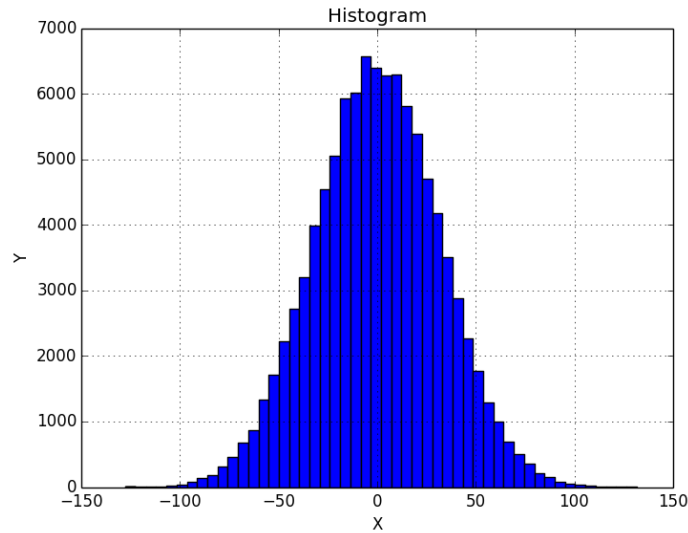


FIG. 2. Histogram of many Brownian paths, demonstrating diffusion.

RANDOM MOTION AND THE WAVE EQUATION

We again follow Zauderer here, using Brownian motion with correlation to model the propagation of a way. The key difference is that the random steps of a particle are not completely independent: the direction at step n is correlated with direction in step $n - 1$. Which is to say, once a particle is in motion, it tends to keep moving in the same direction.

In one dimension, we let p denote the probability that a particle continues in the same direction as the previous step, and q the probability that it reverses direction. To get a continuous motion, one can assume the probability p tends to one as the step size δ decreases to zero, so we can write

$$p = 1 - \lambda\delta + O(\delta^2) \text{ and } q = \lambda\delta + O(\delta^2). \quad (43)$$

If we let $\alpha(x, t)$ represent the density of particles at x that arrived there from the left, and $\beta(x, t)$ the density of particles that arrived from the right, we obtain a system of difference equations

$$\alpha(x, t + \tau) = p\alpha(x - \delta, t) + q\beta(x - \delta, t) \quad (44)$$

$$\beta(x, t + \tau) = p\beta(x + \delta, t) + q\alpha(x + \delta, t). \quad (45)$$

Expanding as Taylor series in δ, τ and setting $c = \delta/\tau$ as the velocity parameter, in the limit as $\delta, \tau \rightarrow 0$ we obtain the coupled system of equations

$$\alpha_t + c\alpha_x = -c\lambda\alpha + c\lambda\beta \quad (46)$$

$$\beta_t - c\beta_x = c\lambda\alpha - c\lambda\beta. \quad (47)$$

This system of PDEs is solvable in this form, but it is convenient to recast it as the usual two-way wave equation.

A particle at point x had to get there either from the left or the right, so the density of particles is given by the sum

$$v(x, t) = \alpha(x, t) + \beta(x, t). \quad (48)$$

Adding equations 46 and 47 gives

$$(\alpha + \beta)_t + c(\alpha - \beta)_x = 0 \quad (49)$$

while subtracting the two gives

$$(\alpha - \beta)_t + c(\alpha + \beta)_x = -2c\lambda(\alpha - \beta). \quad (50)$$

Differentiating equation 49 w.r.t. t and equation 50 w.r.t. x and subtracting gives a differential equation for $v = \alpha + \beta$ as

$$v_{tt} - c^2 v_{xx} + 2\lambda c v_t = 0. \quad (51)$$

This is a special case of the *telegrapher's equation* and is simply the 1D wave equation with a damping term $2\lambda v_t$.

To obtain the Green's function, we start all the random particles at position $x = 0$ at time $t = 0$, with random direction of travel. Since half of them will travel left, the other half to the right, this corresponds to an initial condition for α, β as

$$\alpha(x, 0) = \beta(x, 0) = \frac{1}{2}\delta(x), \quad (52)$$

where here, $\delta(x)$ is the Dirac delta function. This corresponds to the initial conditions for v as

$$v(x, 0) = \delta(x) \text{ and } v_t(x, 0) = 0 \quad (53)$$

since v is the sum of α and β while the time derivative v_t is proportional to the difference $\alpha_x - \beta_x$ by equation 49. In this case, the random particle motion will recover the Green's function $G^{(0)}(x, t)$ which we expect to be two Dirac delta functions travelling at velocity c . Plus, of course, some residual due to the damping term.

NUMERICAL VERIFICATION: 1D WAVE EQUATION

We do a simple numerical example to demonstrate that the random particle motion does indeed create a Green's function for the wave equation.

In this example, we begin with the 1D telegrapher's equation

$$v_{tt} - c^2 v_{xx} + 2\lambda c v_t = 0 \quad (54)$$

which, as mentioned in the previous section, is the wave equation with an additional damping term. We again will write code for random motion, with a large number of random particles all starting at position $x = 0$ at time $t = 0$. For the differential equation, fix the velocity to be $c = 3000$ m/s, the time for the simulation to be $t = 2$ s and set a small value for $\lambda = .0001$ in this example. For the simulation, we choose the number of particles to be $n_{particles} = 10000$, spatial step size $\delta = 0.1$, time step size $\tau = \delta/c$, and probability of travelling in the same direction to be $p = 1 - \lambda\delta = .99999$. The main loop in the code is given as follows:

```

x = zeros(nparticles)
xstep = 2*rand(0:1,nparticles)-1
for k=1:nsteps
    x = x + dx*xstep
    xstep = xstep.*(2*(rand(nparticles).<p)-1)
end

```

where the line “`xstep = xstep.*(2*(rand(nparticles).<p)-1)`” is what forces the particles to tend to move in the same direction at each step.

Running the code, we obtain the histogram shown in Figure 3 representing the density of particles, and we see two strongly focused delta-type peaks at $x = \pm 6000$, which is what we would expect for waves travelling at velocity 3000 m/s over a period of 2 seconds.

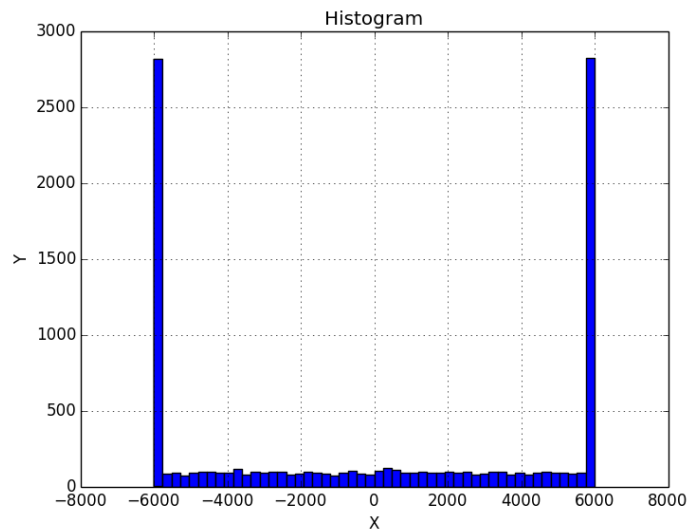


FIG. 3. Histogram of many correlated Brownian paths, giving Green’s function for wave equation.

NUMERICAL EXPERIMENTS: 2D WAVE EQUATION

In the 2D case, we don’t have the benefit of a mathematical proof of random particles simulating a wave equation. Nevertheless, we will try a few experiments.

In the first experiment, we track a single particle as it moves at random in 2 dimension. The particle will have a position in x, y stored in array as components $x[1], x[2]$ and will have a direction θ as the polar angle, stored as component $x[3]$. Starting the particle at the origin $(0, 0)$ and headed in direction of 45° , we will take 10,000 steps of length $\delta = 0.1$ in the following loop:

```

x[:,1] = [0,0,.125]
for k=1:10000
    x[1,k+1] = x[1,k]+0.1*cos(2*pi*x[3,k]) # x_1 component
    x[2,k+1] = x[2,k]+0.1*sin(2*pi*x[3,k]) # yx_2 component
    x[3,k+1] = mod(x[3,k]+.01*(randn()),1)
end

```

end

The sample random path is displayed in Figure 4.

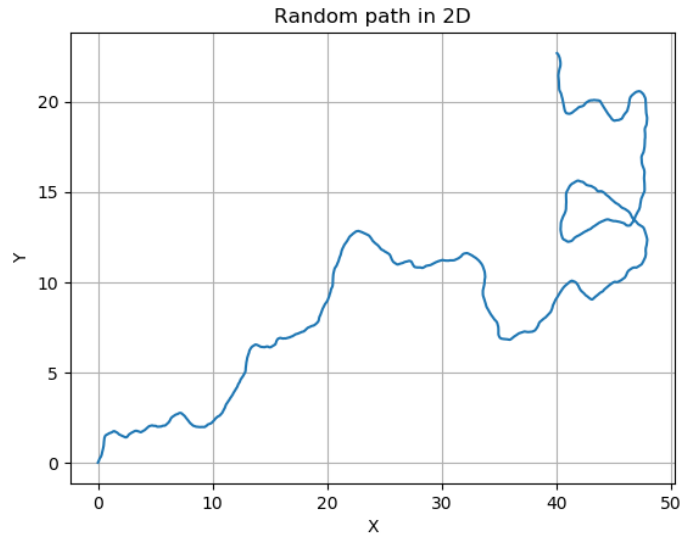


FIG. 4. Brownian motion in 2D, with correlation of direction between adjacent steps.

Changing the correlation parameter `.01` in the `x[3, k + 1]` assignment (of the code) to a smaller value (say `0.003`) results in a smoother, less random path, as shown in Figure 5.

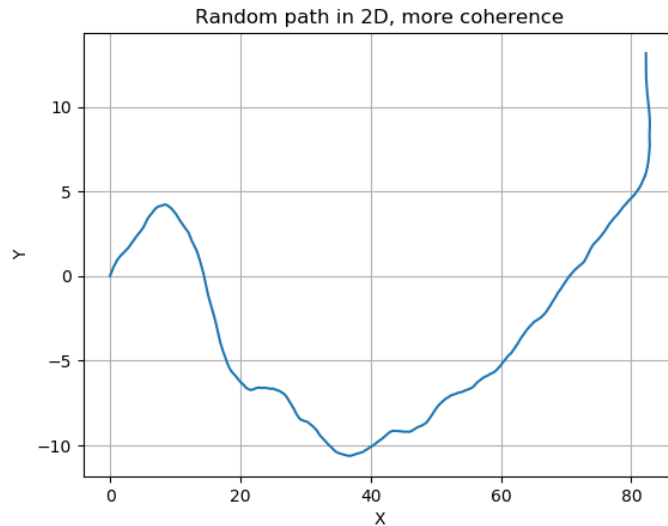


FIG. 5. Brownian motion in 2D, with stronger correlation of direction between adjacent steps.

We now take a simulation with many particles (here, the number of particles is set to `nparticles = 10,000`) all moving at random in two dimensions. The key inner loop in the code updates the `x[1], x[2]` position by moving a distance `dx` in the direction given by the `x[3]` parameter. The `x[3]` parameter (direction) is then randomly changed by a small increment. This inner loop is as follows:

```

for k=1:nsteps
  for j=1:nparticles
    x[1,j] += dx*cos(2*pi*x[3,j])
    x[2,j] += dx*sin(2*pi*x[3,j])
    x[3,j] = mod(x[3,j]+q*randn(),1)
  end
end
end

```

The resulting distribution of particles appears to be a circle of radius 6000 m (2 seconds at speed 3000 m/s), as shown in Figure 6, which is a bit reassuring. However, we know that in 2D, the Green’s function is not limited to a thin set along a circle – there should be a “tail.” So, we compute a more precise histogram of the distribution of the particles at time $t = 2$ s, as a function of radial distance. Figure 7 shows the distribution, and clearly we see there is a tail in place – so not all the particles are concentrated on the circle.

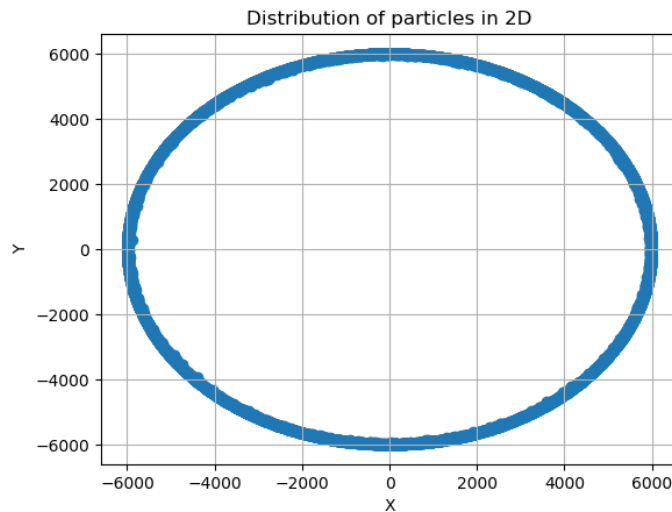


FIG. 6. Distribution of particles 2D, after travelling 6000 m.

It is somewhat comforting to compare the histogram in Figure 7 to a plot of the 2D Green’s function as shown in Figure 8. The shapes are the same – although there is a lot of work to be done to get these to match.

FIRST ORDER HYPERBOLIC SYSTEMS

It is useful to recast the second order linear wave equation as a system of first order partial differential equations, and consider how this might be modelled by a particle simulation.

One obtains a PDE system for the acoustic wave equation by considering two functions $P(x, t)$ and $U(x, t)$ representing acoustic pressure and fluid flow. With ρ the density of the fluid, conservation of mass gives the equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho U)}{\partial x} = 0, \quad (55)$$

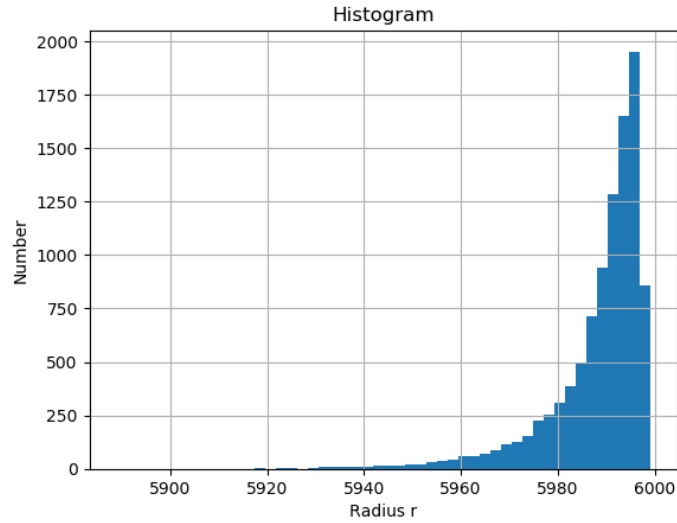


FIG. 7. Histogram of the particle distribution, as a function of radial distance r .

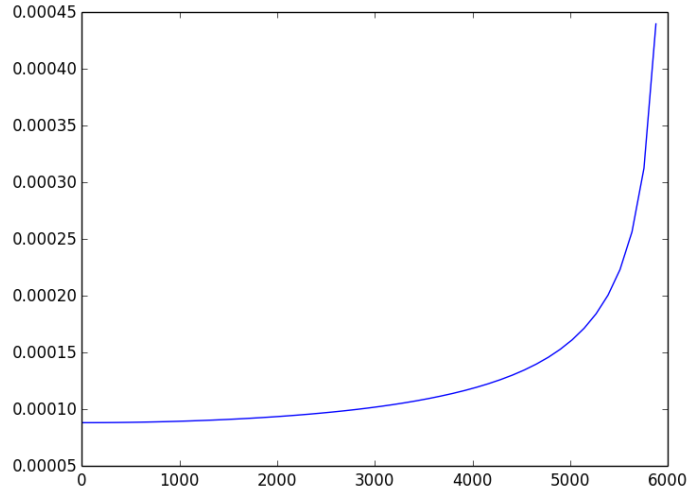


FIG. 8. Histogram of the particle distribution, as a function of radial distance r .

and linearizing density $\rho = \rho_0 + \rho_0 s$ about a median density ρ_0 and applying an equation of state $P = Bs$, we obtain

$$\frac{\partial P}{\partial t} + B \frac{\partial U}{\partial x} = 0. \quad (56)$$

Conservation of momentum, linearized, gives the equation

$$\rho_0 \frac{\partial U}{\partial t} + \frac{\partial P}{\partial x} = 0. \quad (57)$$

Taking these last two equations, we obtain a first order system of two coupled PDEs,

$$\frac{\partial}{\partial t} \begin{pmatrix} P \\ U \end{pmatrix} = \begin{pmatrix} 0 & -B \\ -\frac{1}{\rho_0} & 0 \end{pmatrix} \frac{\partial}{\partial x} \begin{pmatrix} P \\ U \end{pmatrix}. \quad (58)$$

To model this first order system, we consider a swarm of particles made up of two types: type P and type U . These particles will each have a sign attached to them, a ± 1 .

We step in time with step size τ and step in space with step size δ . A particle of type P at x will generate a particle of type U in the next time step, at location $x + \delta$, with probability p' . It will also generate a particle of type $-U$ in the next time step, at location $x - \delta$, with probability q' . Similarly, a particle of type U will generate particles of type $\pm P$ to the right and left of its position in the next time step.

Now let $v(x, t)$ equal the sum of particles of type P at point x , at time t , where the sum takes into account the sign of the particles, and $w(x, t)$ the corresponding sum for the particles of type U . We now have difference equations for v, w that tells us how many particles are present in the next time step. Namely,

$$v(x, t + \tau) = v(x, t) + p''w(x - \delta, t) - q''w(x + \delta, t) \quad (59)$$

and

$$w(x, t + \tau) = w(x, t) + p'v(x - \delta, t) - q'v(x + \delta, t). \quad (60)$$

Expanding in a Taylor series in δ, τ and choosing $p' = q', p'' = q''$ for a cancellation, we obtain the linearized system

$$v_t(x, t)\tau = -2p''w_x(x, t)\delta \quad (61)$$

$$w_t(x, t)\tau = -2p'v_x(x, t)\delta. \quad (62)$$

Choosing our ratio δ/τ and probabilities p', p'' appropriately, in the limit as $\delta, \tau \rightarrow 0$ we obtain the desired system of PDEs,

$$v_t = -Bw_x \quad (63)$$

$$w_t = -(1/\rho_0)v_x. \quad (64)$$

From this we conclude that a simulation with particles following the generating process above will result in a numerical solution to the first order PDE system. Note however, that this is an example of a branching process, as new particles are potentially created at each time step. This may lead to some computational difficulties, as the number of particles will be expected to increase over the duration of the simulation.

FIRST ORDER HYPERBOLIC SYSTEMS - 3D

In three dimensions, the corresponding (normalized) system of coupled PDEs is given as

$$\frac{\partial}{\partial t} \begin{pmatrix} P \\ u^1 \\ u^2 \\ u^3 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{\partial}{\partial x} & -\frac{\partial}{\partial y} & -\frac{\partial}{\partial z} \\ -\frac{\partial}{\partial x} & 0 & 0 & 0 \\ -\frac{\partial}{\partial y} & 0 & 0 & 0 \\ -\frac{\partial}{\partial z} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} P \\ u^1 \\ u^2 \\ u^3 \end{pmatrix}, \quad (65)$$

where P is the acoustic pressure and $\mathbf{U} = (u^1, u^2, u^3)$ is the fluid flow in three dimensions. It seems we would need to consider four types of particles moving randomly in the particle swarm in order to model this system of PDEs. We have not explored this idea in detail yet.

COMPUTATIONAL SPEED UPS

The goal of this exercise is to get a method that can rapidly compute Green's functions, and consequently calculate wave propagation from source to receivers in a typical seismic simulation. Some features we intend to exploit:

1. Each particle moves independently of the others. Thus the computational burden of following many particles can be distributed across many processors, which do not need to communicate with each other. This is a significant advantage over finite difference methods, where computational cells must communicate with their neighbouring cells.
2. The computations are simple, mainly involving additions and multiplications, thus well suited for graphical processing units (GPUs). A graphics card running CUDA may be a very efficient way of achieving a highly parallel, accelerated computation.
3. There are no computational boundaries. We do not have to worry about implementing numerical absorption of a particle at a computational boundary, as the particles can move without bound, or at least up to the numerical range of IEEE floating point numbers. Indeed, once a particle has travelled too far, we can remove it from the simulation since it will not have time to return to a receiver. Exiting particles free up computational resources for other particles.
4. There is considerable flexibility in choice of parameters: number of particles, step sizes, correlations between steps. We may be able have choices that give both high accuracy as well as speed.

We did some initial experiments with timing, only on a choice of languages. We did observe that Matlab is indeed very fast, and faster than our current favourite, the programming language Julia. In highly optimized code, we performed a tight loop of 10,000 particles in 2D, using 60,000 steps to obtain timings of

- 14.9 seconds in Matlab (3.5 GHz 6-Core Intel Xeon E5)
- 26.5 seconds in Julia (3.5 GHz 6-Core Intel Xeon E5)

From these measurements, it is observed Matlab is about twice as fast.

It is also interesting to note that Matlab ran faster with code written as vectors, while Julia was faster with the vector operations unrolled into a for loop on components.

INHOMOGENEOUS MEDIA

The examples above all assumed constant coefficients for the wave equation, or more generally for the telegrapher's equation. However, as discussed in Zauderer (1989), by choosing the probabilities for the random motion so that change of persistence in motion to

the left is different than the persistence in motion to the right, the 1D differential equation that arise in the limit of particle motion becomes

$$v_{tt} + c^2[\sigma(\sigma v)_x]_x + 2\sqrt{\lambda}c(\sigma\psi v)_x + 2\lambda cv_t = 0, \quad (66)$$

where functions $\sigma = \sigma(x)$, $\psi = \psi(x)$ are functions that characterize these different probabilities. We are skipping the details here, only to point out that inhomogeneous media for wave propagation can in principle be simulated by this particle motion.

SOME PROBLEMS AND POSSIBLE SOLUTIONS

A major problem with this method is that a probability distribution, as modelled by random particles, is always positive, while a Green's function need not be. In particular, if the medium for the waves has a reflective layer, it could induce a change in polarity so a negative Green's function can result there. Also, in 2D and 3D, the first Green's function $G^{(0)}$ takes negative values even in the constant coefficient case.

As suggested in the section on hyperbolic systems, a possible solution may be to include particles with a negative "sign" in the simulation, to produce a signed probability distribution.

A second problem is that our 1D simulation only generates the first Green's function $G^{(0)}$ corresponding to an initial condition of a delta function for $u(x, 0)$, with $u_t(x, 0) = 0$. We do not know (yet) how to compute the Green's function $G^{(1)}$. One challenge is that $G^{(1)}$ is not a normalized probability distribution – its total mass grows linearly with time, which would suggest the number of particles would need to increase with time.

Also as suggested in the section on hyperbolic systems, a possible solution may be to include particles that spawn off new particles, or to allow the weight (or "mass") of the particles to increase. Or perhaps particles move while keeping track of a "tail" that records where their path was.

SUMMARY

Randomly moving particles may be used as a computational model for a numerical simulation of partial differential equations in several spatial variables. We demonstrated examples in one and two spatial dimensions to show how a Green's function for the heat equation, and the acoustic wave equation, can be approximated by a simulation with many random particles. We considered a simple case of a coupled system of first order partial differential equation in one spatial dimension that can be modelled by a branching process of super Brownian particle motion, and showed the limiting density of the particles forms a solution to the PDEs. This system required a model with two types of particles. In three spatial dimension, we expect four types of signed particles to be required.

A few numerical examples were presented to demonstrate the simulations.

FUTURE WORK

This is very much a work in progress. Our goal is to show the randomly moving particles, with no interaction between the particles, can be used to model the wave equation. We expect this to provide significant computation benefits, particularly in parallelization. We will draw on the extensive literature on stochastic modelling to make rigorous the ideas presented in this outline.

ACKNOWLEDGEMENTS

We thank the sponsors of CREWES for their support. We also gratefully acknowledge support from NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 461179-13 and through a Discovery Grant for the first author.

REFERENCES

- Crisan, D., and Lyons, T., 1999, A particle approximation of the solution of the Kushner-Stratonovitch equation: *Probab. Theory Relat. Fields*, **155**, 549–578.
- Kozdron, M. J., 2008, *Brownian motion and the heat equation: Lecture notes for ACSC 456*.
- Lawler, G. F., 2010, *Random walk and the heat equation: American Mathematical Society*.
- Polimeridis, A. G., Yioultsis, T. V., and Tsiboukis, T. D., 2007, Fast numerical computation of Green's functions for unbounded planar stratified media with a finite-difference technique and Gaussian spectral rules: *IEEE Trans. on microwave theory and techniques*, **55**, 1056–1067.
- Quer-Sardanyons, L., and Tindel, S., 2007, The 1-D stochastic wave equation driven by a fractional Brownian motion: *Stochastic Processes and their Applications*, **117**, 1448–1472.
- Slade, G., 2002, Scaling limits and super-Brownian motion: *Notices of the AMS*, **49**, 1056–1067.
- Toomey, A., and Bean, C. J., 2000, Numerical simulation of seismic waves using a discrete particle scheme: *Geophys. J. Int.*, **141**, 595–604.
- Yang, X., and Li, X., 2015, Full-discrete finite element method for stochastic hyperbolic equation: *Journal of Computational Mathematics*, **33**, 533–556.
- Zauderer, E., 1989, *Partial Differential Equations of Applied Mathematics: Wiley Interscience*.